

封装

- ◆ 为什么要使用封装
- ◆ 什么是封装
- ◆ 如何使用封装
- ◆ this的用法

◆为什么要使用封装

▶ 下面代码有什么问题？

```
Dog d = new Dog();  
d.health = 1000;
```

不合理的赋值

▶ 如何解决上面设计的缺陷？

使用**封装**

◆ 什么是封装

▶ 面向对象三大特征之一——封装

▶ 封装的概念

封装：将类的某些信息隐藏在类内部，不允许外部程序直接访问，而是通过该类提供的方法来实现对隐藏信息的操作和访问

▶ 封装的好处

只能通过规定方法访问数据

隐藏类的实现细节

方便加入控制语句

方便修改实现



如何使用封装

封装的步骤

```
class Dog {  
    private String name = "旺财"; // 昵称  
    private int health = 100; // 健康值  
    private int love = 0; // 亲密度  
    private String strain = "拉布拉多犬"; // 品种  
    public int getHealth() {  
        return health;  
    }  
    public void setHealth (int health) {  
        if (health > 100 || health < 0) {  
            this.health = 40;  
            System.out.println("健康值应该在0和100之间,默认值是40");  
        }  
        this.health = health;  
    }  
    // 其它getter/setter方法  
}
```

1

2

3

this代表
当前对象

创建getter/setter方法

用于
读

在getter/setter方法中
加入属性控制语句

对属性值的
合法性进行
判断

Dog

- name:String
- health:int
- love:int
- strain:String

+ print():void
+ setHealth():void
+ getHealth():String
... ..

▶ this关键字的用法

▶ 调用属性

```
this.health = 100;  
this.name = "大黄";
```

▶ 调用方法

```
this.print();
```

▶ 调用构造方法

```
this();
```

```
this("小黑",100,100,"雄");
```

如果使用，必须是构造方法中的第一条语句

◆ 练习一编写Dog和Penguin类

▶ 需求说明：

- ▶ 运用面向对象思想抽象出Dog类和Penguin类

类型	属性				行为
狗	昵称	健康值	亲密度	品种	输出信息
企鹅	昵称	健康值	亲密度	性别	输出信息

- ▶ 编写Dog类和Penguin类
- ▶ 添加默认构造方法

◀练习——打印Dog信息1

- ▶ 根据控制台提示信息选择领养宠物（狗），
 - ▶ 输入昵称、品种、健康值
 - ▶ 打印宠物信息
- ▶ 要保证健康值的有效性（在1到100之间）

◀ 练习——打印Dog信息2

▶ 实现思路：

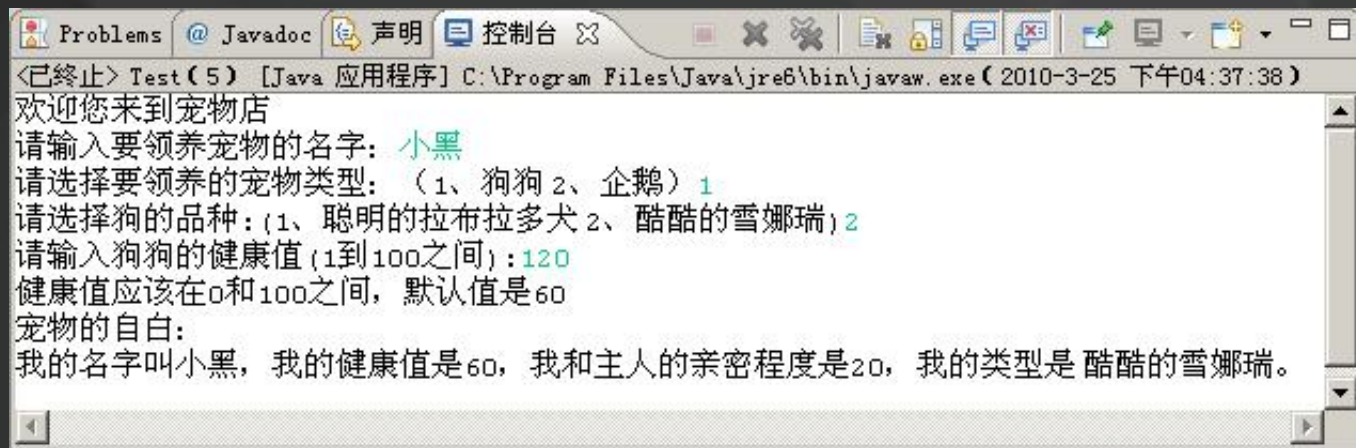
▶ 修改Dog类

保证健康值的有效性，否则取默认值60

▶ 编写Test类

从控制台输入宠物信息

打印宠物信息



```
<已终止> Test(5) [Java 应用程序] C:\Program Files\Java\jre6\bin\javaw.exe (2010-3-25 下午04:37:38)
欢迎您来到宠物店
请输入要领养宠物的名字: 小黑
请选择要领养的宠物类型: (1、狗狗 2、企鹅) 1
请选择狗的品种: (1、聪明的拉布拉多犬 2、酷酷的雪娜瑞) 2
请输入狗狗的健康值 (1到100之间): 120
健康值应该在0和100之间, 默认值是60
宠物的自白:
我的名字叫小黑, 我的健康值是60, 我和主人的亲密程度是20, 我的类型是 酷酷的雪娜瑞。
```

- ▶ 为什么要使用封装
- ▶ 如何实现封装

谢 谢

